CPU

D. Leeuw

14 maart 2024
v.0.8.1

Over dit Document

0.1 Voorkennis

Voor het doorgronden van de in dit document behandelde stof is er geen specifieke voorkennis vereist. Het is wel makkelijk als de lezer weet wat binair is.

0.2 Leerdoelen

Na het bestuderen van dit document heeft de lezer een globale kennis van de werking van de centrale processing eenheid (CPU) volgens het model van Von Neumann.

Inhoudsopgave

Hoofdstuk 1

CPU

De computer heet een computer omdat hij kan rekenen (to compute), maar de echte rekenaar van de computer is de CPU de rest van de computer is er alleen om ons, gebruikers, te ondersteunen. De functie van de CPU is de verwerking van instructies en data. Het doet dit door een enorme hoeveelheid transistoren die geïntergreerd zijn op een stuk gezuiverd silicium (silicon). Het geheel wordt geproduceerd op een wafer, wat een grote ronde plaat is met vele toekomstige CPU's erop. De wafer wordt in stujes gebroken tot de zogenaamde "dies". Elke "die" wordt verplakt in een behuizen en verkocht als de CPU of processor.

Moderne processoren zijn complexe chips waarop veel meer geïntegreerd is dan alleen de rekeneenheid (ALU). Het is een samenraapsel van allerlei functies die samen op een stuk silicium (Engels: die) zijn samengebracht. De reden dat het een samenraapsel is is omdat er in de loop van de tijd steeds meer functies van het moederbord zijn overgebracht naar de processor omdat dat het geheel sneller maakt.

1.1 Clock

Een computer heeft net als een mens een hart nodig. Het ritme van het computerhart bepaalt de snelheid waarop de computer kan werken. Het hart in de computer is een kristal. Als je een spanning op een kristal zet gaat deze in een bepaalde frequentie trillen. Deze frequentie wordt gebruikt als heartbeat in de computer. In de computer is de heartbeat een blokgolf.

Via chips en gebruik makend van de opgaande en neergaande flank van de clock frequentie kunnen er binnen de computer verschillende snelheden gebruikt worden om bepaalde processen aan te sturen.

1.2 ALU

De Arithmetic Logical Unit is één van de kloppende harten van een CPU. Het is de feitelijke verwerkingsunit. De ALU kent drie inputs:

- Operand A
- Operand B
- Uit te voeren instructie (I)

Daarnaast heeft een ALU twee outputs:

- Het resultaat van A instructie B (R)
- De status (S) van de uitgevoerde instructie; goed of fout

Een functie kan een rekenkundige functie zijn zoals optellen of vermenigvuldigen, maar het kan ook een logische functie zijn zoals een AND of een XOR.

Een ALU kan alleen met gehele getallen (integers) werken, niet met getallen waarin een komma voorkomt (floating point).

1.3 FPU

De Floating Point Unit is de rekeneenheid die het rekenen met decimalen voor zijn rekening neemt. De FPU is een unit die niet altijd op de die heeft gezeten. In oudere machines zat er een apparte chip op het moederbord die de FPU was (voor Intel since de i486 op de die).

1.4 CU

De Control Unit is de eenheid die binnen de CPU zorg draagt voor de timing en stuursignalen. De CU is o.a. verantwoordelijk voor de communicatie tussen de CPU en de rest van de wereld. De hoofdtaak van de CU is het doorlopen van het volgende proces:

1. Fetch the instruction
2. Fetch the operands
3. Decode the instruction (naar ALU of FPU)
4. Execute the instruction

5. Als de status ok is, schrijf het resultaat weg naar het geheugen
6. Als de status niet ok is, handel de error af

De CPU kent vele verschillende instructies mede afhankelijk van het type processor en de leverancier. De meeste basis instructies die bijna alle processoren ondersteunen zijn:

Calculation add, sub, inc, dec, mul, div

Logical operators and, or, xor

Other mov, loop, jump

1.5 Registers

Registers zijn stukken geheugen dicht op de ALU en FPU. In de ALU registers komen bijvoorbeelde de operants en de instructie te staan en na een clock-tik komen het resultaat en de status in een ander register te staan. Deze General Purpose registers is het geheugen waarmee de ALU werkt.

Hoofdstuk 2

De CU en de wereld

2.1 Bussen

De CPU is gekoppeld met drie bussen aan de rest van de wereld. De drie bussen zijn:

Address bus Selecteert een adres in het geheugen, de enen en nullen op de draden bepalen welk adres geselecteerd is. Is bijvoorbeeld 32, 36, 48 of 64 bits breed en bepaalt daarmee de maximale hoeveelheid geheugen die kan worden aangesproken.

Control bus Geeft aan wat er met de data op het adres moet gebeuren (bijvoorbeeld lezen of schrijven). Bijvoorbeeld 8 bits, afhankelijk van de hoeveelheid instructies die een processor moet ondersteunen.

Data bus Bus waarover de data getransporteerd wordt. Is bijvoorbeeld 32 of 64 bits breed en geeft aan wat voor architectuur we hebben.

Systeem	Databus	Adresbus	Max geheugen
32 bits	32 bits	32 bits	4 GB
		36 bits	64 GB
64 bits	64 bits	48 bits	256 TB
		64 bits	16 EB

2.2 Cache

De CPU kent verschillende soorten cache:

- De L1 (level 1) cache is cache het dichtst op de rekeneenheden (ALU en FPU). Deze cache bestaat uit een cache voor instructies en een cache

voor data. Dit is na de General Purpose registers het snelste geheugen, maar er is het minst van aanwezig.

- Er is ook een L2 cache. Dit is een gedeelde cache voor data en instructies. Is groter dan L1 cache, maar ook trager.
- Tot slot is er nog een L3 cache die gedeeld wordt tussen de verschillende cores in een CPU.

De reden voor de toevoeging van cache aan het systeem is gebeurd omdat de CPU steeds sneller werd en de snelheid van het RAM daarin achterbleef. Cache is sneller dan RAM en dient daarom als buffer tussen de CPU en het RAM-geheugen.

Hoofdstuk 3

CPU optimalisatie

3.1 Overclocking

Het is mogelijk om een CPU op een hogere snelheid te laten werken dan waarvoor de fabrikant hem verkocht heeft. CPU's worden gemaakt op gezuiverd silicium, maar dat is nooit 100% zuiver. Kleine verontreinigingen zorgen voor kleine verschillen tussen de CPU's van één wafer. De fabrikant test de verschillende dies en beslist dan voor welke clock snelheid de die verkocht kan worden. Over het algemeen is dat de snelheid waarbij er voor 100% zekerheid gesteld kan worden dat er geen fouten optreden. Daar zit natuurlijk een veiligheidsmarge op. De fabrikant wil geen claim aan zijn broek krijgen van een bank dat er een miljarden overschrijving fout is gegaan.

Voor thuis kan een iets hogere snelheid met het risico op kleine foutjes echter wel acceptabel zijn. Dit op een hogere snelheid laten werken, dan waarvoor de CPU verkocht werd, heet overclocking. Er zijn verschillende manieren waarop overclocking bereikt kan worden:

Core speed overclocking Zorgen dat de snelheid waarmee de CPU rekent (ALU, FPU) omhoog gaat

FSB overclocking Zorgen dat de data sneller van en naar het RAM gaat

beide Gebruik van beide technieken

Als de CPU meer berekeningen per seconde moet doen, moet hij meer arbeid verzetten en dus meer vermogen verbruiken wat ook betekent dat hij meer warmte produceerd. We zullen dus beter moeten koelen en rekening houden met extra elektriciteit verbruik.

3.2 CPU Throtteling

Veel CPU's in moderne machines hoeven niet continue hun volle vermogen te leveren, sterker de meeste CPU's staan het merendeel van hun tijd te wachten op instructies. Dat is natuurlijk zonde van het verbruikte vermogen. Het zou beter zijn dat als de CPU niets te doen heeft hij ook minder vermogen verbruikt. De simpelste manier is het omlaag brengen van de clock snelheid. Dit proces heet CPU throtteling.

Bij komende voordelen van CPU throtteling is verlaging van de temperatuur en het verlengen van de batterij duur.

3.3 Cores

Er bestaat de mogelijkheid om meerdere CPU's op een moederbord te plaatsen. Dit heet Symmetric Multiprocessing (SMP). Het besturingssysteem moet SMP aware zijn, het ondersteunen, om gebruik te kunnen maken van de meerdere processoren die aanwezig zijn.

Om het moederbord minder complex te maken kunnen er ook meerdere dies in een behuizing gestopt worden. Je hebt dan meerdere cores in een CPU, ook dit is SMP en moet het OS weten hoe hiermee om te gaan. Het staat bekent als Chip Level Multiprocessing (CLM) en kan bestaan uit 2, 4, 8 of meerdere cores in een enkele CPU behuizing.

Verdere integratie vinden we bij meerdere cores op een die.

3.4 Pipeline optimalisatie

Een core handelt instructies één voor één af. Als we een simpel stappen plan (thread, pipeline) nemen:

1. Fetch operands
2. Fetch instruction
3. Decode instruction
4. Execute instruction
5. Write output of instruction to RAM

Dan kunnen we per clock tick een stap nemen. We kunnen ook per clock tick twee stappen doen. We kunnen namelijk bij de opgaande flank van de clock een instructie doen en één bij de neergaande flank. Dit heet superpipelining.

Een stap bij de opgaande en bij de neergaande flank heet een sub-stage. Er zitten dus 2 sub-stages in 1 clock cycle. Met sub-stage maken we de processor twee keer zo snel.

Als we deze processen parallel doen in meerdere cores kunnen we spreken van een Superscalar Architecture.

Met voldoende registers kunnen we nog meer stappen tegelijk doen:

Clock tick 1 opgaande flank Haal operand A1 uit RAM en plaats in register A (adresbus, databus en controlbus bezet, register A bezet)

Clock tick 1 neergaande flank Haal operand B1 op uit RAM en plaats in register B (adresbus, databus en controlbus bezet, register A en B bezet)

Clock tick 2 opgaande flank Haal de instructie 1 op uit RAM en plaats in register C (adresbus, databus en controlbus bezet, register A, B en C bezet)

Clock tick 2 neergaande flank Decodeer instructie 1 en haal operand A2 op uit RAM en plaats in register D (adresbus, databuse en controlbus bezet, register A,B,C en D bezet)

clock tick 3 opgaande flank Voer instructie 1 uit en plaats resultaat en de status in registers E en F en we halen operand A2 op uit RAM en plaatsen het in register G (adresbus, databus en controlbus bezet, register D,E,F en G bezet)

clock tick 3 neergaande flank Haal instructie 2 op uit RAM en plaats in register C (adresbus, databus en controlbus bezet, register C,D,E,F en G bezet)

clock tick 4 opgaande flank Decode instructie 2 en write output instructie 1 naar RAM (adresbus, databus en controlbus bezet, register C,D en G bezet)

Clock tick 4 neergaande flank Voer instructie 2 uit op registers D en G en stop resultaat en status in A en B, haal operand A3 op uit RAM en plaats in E (adresbus, databus en controlbus bezet, registers A,B en E bezet)

etc.

Op deze manier kunnen er meerdere instructies tegelijk uitgevoerd worden. Binnen een externe clock tick gebeurt er in de CPU meer dan je bij 1 clock

tick zou verwachten. Ook kunnen opdrachten misschien wel vast uitgevoerd worden op de FPU terwijl de ALU bezig is, zo kan er nog veel meer parallel.

Met meer cores kunnen we threads of pipelines ook nog eens verdelen over de cores zodat we kunnen spreken over Hyper-threading (HT of HTT) bij Intel CPUs of over Simultaneous Multithreading (SMT) bij AMD CPUs.

Software moet instaat zijn om met HTT of SMT om te gaan. Vaak kan de techniek in de BIOS uit gezet worden. Het voordeel is dat de CPU minder idle is, maar er zijn meer registers nodig. Er zijn hacks bekend die gebruik maken van HTT om security informatie buit te maken, dus soms wordt er geadviseerd om HTT uit te zetten.

Hoofdstuk 4

Opdrachten

4.1 Verzamel CPU informatie vanuit het OS

Windows In PowerShell gebruik het `Get-WmiObject` commando om de processor gegevens op te vragen.

Mac OS X Gebruik `sysctl` om te achterhalen welke processor er in je Macintosh zit.

Linux gebruik `dmidecode` om de gegevens van je CPU te achterhalen.